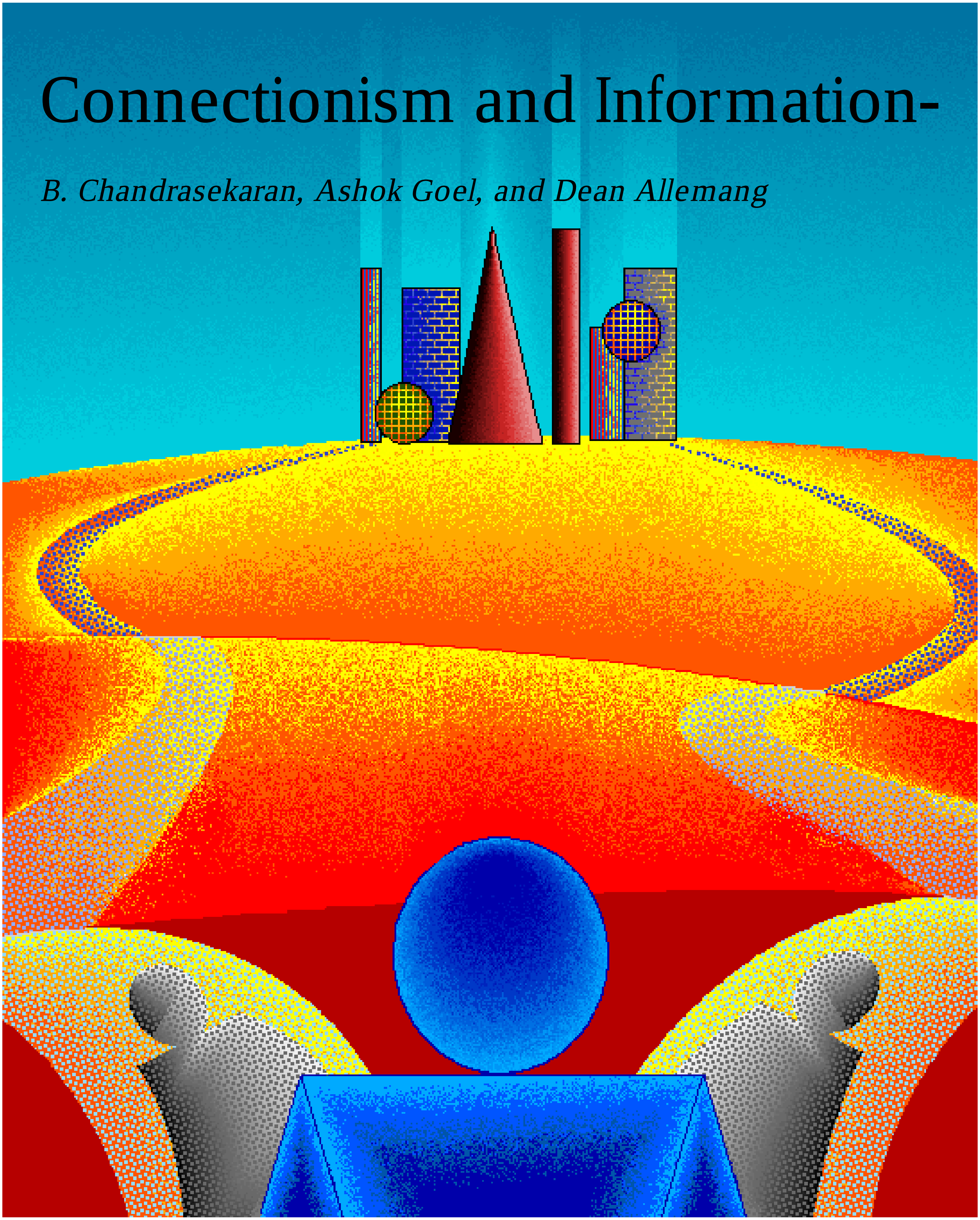
AI Magazine Volume 9 Number 4 (1988) (© AAAI)

# Connectionism and Information-

*B. Chandrasekaran, Ashok Goel, and Dean Allemang*



# Processing Abstractions

## The Message Still Counts More Than the Medium

*Connectionism challenges a basic assumption of much of AI, that mental processes are best viewed as algorithmic symbol manipulations. Connectionism replaces symbol structures with distributed representations in the form of weights between units. For problems close to the architecture of the underlying machines, connectionist and symbolic approaches can make different representational commitments for a task and, thus, can constitute different theories. For complex problems, however, the power of a system comes more from the content of the representations than the medium in which the representations reside. The connectionist hope of using learning to obviate explicit specification of this content is undermined by the problem of programming appropriate initial connectionist architectures so that they can in fact learn. In essence, although connectionism is a useful corrective to the view of mind as a Turing machine, for most of the central issues of intelligence, connectionism is only marginally relevant.*

### Challenge to the Symbolic View

Much of the theoretical and empirical research in AI over the past 30 years has been based on the so-called symbolic paradigm—the thesis that algorithmic processes which interpret discrete symbol systems provide a good basis for modeling human cognition. Stronger versions of the symbolic paradigm were proposed by Newell (1980) and Pylyshyn (1984). Newell's physical symbol system hypothesis is an example of the symbolic view. Pylyshyn argues that symbolicism is not simply a metaphoric language to talk about cognition but that cognition literally is computation over symbol systems. It is important to note that the symbolic view does not imply a restriction to serial computation or a belief in the practical sufficiency of current von Neuman computer architectures for the task of understanding intelligence. Often, disagreements about symbolicism turn out to be arguments for computer architectures that support some form of parallel and distributed processing rather than arguments against computations on discrete symbolic representations.

In spite of what one might regard as significant AI achievements in providing a computational language to talk about cognition, recurring challenges have been made to the symbolic paradigm. A number of alternatives have been proposed whose basic mechanisms are not in the symbol-interpretation mode. Connectionism is one such alternative. It revives the basic intuitions behind the early perceptron theory (Rosenblatt 1962) and offers largely continuous, nonsymbol-interpreting processes as a basis for modeling human cognition and perception.

Connectionism and symbolicism both agree on the idea of intelligence as information processing of representations but disagree about the medium in which the representations reside and the corresponding processing mechanisms. We believe that symbolicism and connectionism carry a large amount of unanalyzed assumptional baggage. For example, it is not clear if many of the theories cast in the symbolic mode really require this form of computation and what role the connectionist architecture plays in a successful connectionist solution to a problem. We examine the assumptions and the claims of connectionism in this article to better understand the nature of representations and information processing in general.

### The Nature of Representations: Roots of the Debate

The symbolic versus connectionist debate in AI today is the latest version of a fairly classic contention between two sets of intuitions, each leading to a *weltanschauung* about the nature of intelligence. The debate can be traced in modern times at least as far back as Descartes (to Plato if one wants to go further back) and the mind-brain dualism known as Cartesianism. In the Cartesian world view, the phenomena of the mind are exemplified by language and thought. These phenomena can be implemented by the brain but are seen to have a constituent structure in their own terms and can be studied abstractly. Symbolic logic and other symbolic representations are often advanced as the appropriate tools for studying these phenomena.

*This article is an expanded version of part of a larger paper entitled "What Kind of Information Processing Is Intelligence? A Perspective on AI Paradigms and a Proposal" that appears in* Foundations of Artificial Intelligence: A Source Book, *eds. Derek Partridge and Yorick Wilks, Cambridge University Press, 1988. A much abridged version appears in* Brain and Behavioral Sciences *11(1) as a commentary on a paper on connectionism.*

Functionalism in philosophy, information-processing theories in psychology, and the symbolic paradigm in AI all share these assumptions. Although most of the intuitions that drive this point of view arise from a study of cognitive phenomena, the thesis is often extended to include perception; for example, for Bruner (1957), perception is inference. In its modern version, the Cartesian viewpoint appeals to the Turing-Church hypothesis as a justification for limiting attention to symbolic models. These models ought to suffice, the argument goes, because even continuous functions can be computed to arbitrary precision by a Turing machine.

The opposing view springs from skepticism about the separation of the mental from the brain-level phenomena. The impulse behind anti-Cartesianism appears to be a reluctance to assign any kind of ontological independence to the mind. In this view, the brain is nothing like the symbolic processor of Cartesianism. Instead of what is seen as the sequential and combinational perspective of the symbolic paradigm, some of the theories in this school embrace parallel, holistic (that is, they cannot be explained as compositions of parts), nonsymbolic alternatives; however, others do not even subscribe to any kind of information processing or representational language in talking about mental phenomena. Those who do accept the need for information processing of some type nevertheless reject processing of labeled symbols and look to analog, or continuous, processes as the natural medium for modeling the relevant phenomena. In contrast to Cartesian theories, most of the concrete work deals with perceptual and motor phenomena, but the framework is meant to cover complex cognitive phenomena as well.

Eliminative materialism in philosophy, Gibsonian theories in psychology, and connectionism in psychology and AI can all be grouped as more or less sharing this perspective, even though they differ from each other on a number of issues. The Gibsonian direct perception theory (Gibson 1950), for example, is nonrepresentational. Perception, in this view, is neither an inference nor a product of any kind of information processing; rather, it is a one-step mapping from stimuli to categories of perception made possible by the inherent properties of the perceptual architecture. All the needed distinctions are already directly in the architecture, and no processing over representations is needed.

We note that the proponents of the symbolic paradigm can be happy with the proposition that mental phenomena are implemented by the brain, which might or might not have a symbolic account. However, the anti-Cartesian theorists cannot accept this duality. They want to show the mind as epiphenomenal. To put it simply, the brain is all there is, and it isn't a computer.

Few people in either camp subscribe to all the features in our descriptions. Connectionism is a less radical member of the anti-Cartesian camp because many connectionists do not have any commitment to brain-level theory making. Connectionism is also explicitly representational—its main argument is only about the medium of representation. The purpose of the preceding account is to help in understanding the philosophical impulse behind connectionism and the rather diverse collection of bedfellows that it has attracted.

## Symbolic and NonSymbolic Representations

To better understand the difference between the symbolic and nonsymbolic approaches, let us consider the problem of multiplying two positive integers. We are all familiar with algorithms to perform this task. We also know how the traditional slide rule can be used to do this multiplication. The multiplicands are represented by their logarithms on a linear scale, which are then added by being set next to each other; the result is obtained by reading off the sum's antilogarithm. Although both the algorithmic and slide rule solutions are representational, in no sense can either of them be thought of as an implementation of the other. They make different commitments about what is represented. Striking differences also exist between them in computational terms. As the size of the multiplicands increases, the algorithmic solution suffers in the amount of time it takes to complete the solution, and the slide rule solution suffers in the amount of precision it can deliver.

Let us call the algorithmic and slide rule solutions $S_1$ and $S_{2.}$ Consider another solution, $S_{3,}$ which is the simulation of $S_2$ by an algorithm. $S_3$ can simulate $S_2$ to any desired accuracy. However, $S_3$ has radically different properties from $S_1$ in terms of the information that it represents. $S_3$ is closer to $S_2$ representationally. Its symbol-manipulation character is at a lower level of abstraction altogether. Given a black-box multiplier, ascription of $S_1$or $S_2$ (among others) about what is really going on results in different theories about the process. Each theory makes different representational commitments. Further, although $S_2$ is analog, the existence of $S_3$ implies that the essential characteristic of $S_2$ is not continuity but a radically different sense of representation and processing than $S_1$.

The connectionist models relate to the symbolic models in the same way $S_2$ relates to $S_{1.}$ An adequate discussion of what makes a symbol requires more space and time than we currently have (Pylyshyn [1984] provides a thorough and illuminating discussion), but the following points are useful. A type-token distinction exists: Symbols are types about which abstract rules of behavior are known and can be brought into play. This distinction leads to symbols being labels that are interpreted during the process; however, no such interpretations exist in the process of slide rule multiplication (except for input and output). Thus, the symbol system can represent abstract forms, and $S_2$ performs its addition or multiplication not by instantiating an abstract form, but by having, in some sense, all the additions and multiplications directly in its architecture.

Although we use the word "process" to describe both $S_1$ and $S_{2,}$ strictly speaking no process exists in the sense of a temporally evolving behavior in $S_{2.}$ The architecture directly produces the solution. This is the intuition present in Gibson's (1950) theory of direct perception as opposed to Bruner's (1957) alternative proposal of perception as inference, because the process of inference implies a temporal sequence. Connectionist systems can have a temporal evolution, but unlike algorithms, information processing does not have a step-by-step character. Thus, connectionist models are often presented as holistic.

The main point of this subsection is that functions exist for which the symbolic and connectionist accounts can differ fundamentally in terms of the representational commitments they make. Having granted that connectionism can make a theoretical difference, we now want to argue that the difference connectionism makes is relatively small to the practice of most AI as a research enterprise. Although our arguments refer specifically to connectionist models, they are actually intended to apply to nonsymbolic theories in general.

### Three Ways to Multiply Numbers

Consider the problem of multiplying 45 x 17 to get 765. A classical algorithmic approach to this problem is to do it the way we were taught in school, showing the work:

45<br>17<br>315<br>45<br>765

However, we can also use a slide rule. On a slide rule, the number 45 is written log(45) units from the end of the rule. Hence, to multiply 45 x 17, we line up the distances log(45) and log(17) next to each other, which gives us the place on the rule at log(45) + log(17) = log(765), which is labeled 765, the desired answer (see figure 1).

Notice that if we use the pencil-and-paper algorithm on larger numbers, we use more pencil lead and spend more time writing, and that if we use the slide rule, the answer is less precise.

In the pencil-and-paper example, we are dealing with integer multiples of powers of ten and using the columns to keep track of symbolic representations of them. In the case of the slide rule, we are dealing with logarithms and letting the architecture of the slide rule keep track of them.

We can also solve the multiplication by simulating the slide rule with a computer. That is, we can compute the logarithms to any desired accuracy, add them up, to get the logarithm of the answer.

log 45 + log 17 = log 765
1.653 + 1.230 = 2.883

In this solution, the objects are still logarithms, but the addition is done symbolically. Just as with the slide rule, when the numbers get larger, the answer is less precise. The interesting characteristics of each solution come from the representational commitments it makes, not from the symbolic-nonsymbolic nature of its architecture.

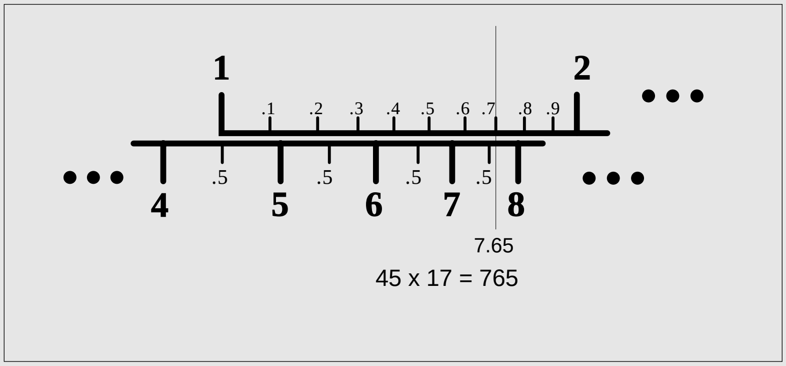


*Figure 1. Multiplication Using a Slide Rule.*

## Connectionism and Its Main Features

Connectionism as an AI theory comes in many different forms. Exactly what constitutes the essence of connectionism is open to debate. The connectionist architectures in the perceptron/parallel distributive processing style (Rosenblatt 1962; Rumelhart et al. 1986) share the following ideas. The representation of information is in the form of weights of connections between processing units in a network, and information processing consists of the units transforming their input into some output, which is then modulated by the weights of connections as input to other units. Connectionist theories emphasize a form of learning in which the weights are adjusted continuously so that the network's output tends toward the desired output. Although this description is couched in nonalgorithmic terms, in fact, many connectionist theorists describe the units in their systems in terms of algorithms that map their input into discrete states. However, the discrete-state description of the units' output, as well as the algorithmic specification of the units' behavior in a connectionist network, is largely irrelevant. This approach is consistent with Smolensky's (1988) statement that the language of differential equations is appropriate to use when describing the behavior of connectionist networks. Further, although our description is couched in the form of continuous functions, the essential aspect of the connectionist architecture is not the property of continuity; it is that the representation medium has no internal labels that are interpreted and no abstract forms that are instantiated during processing. Sidebar 2, entitled A Connectionist Solution to Word Recognition, describes a specific connectionist proposal for word recognition.

A number of properties of such connectionist networks are worthy of note and explain why connectionism is viewed as an attractive alternative to the symbolic paradigm. first is parallelism. Although theories in the symbolic paradigm are not restricted to serial algorithms, connectionist models are intrinsically parallel. Second is distribution. In some connectionist schemes (McClelland, Rumelhart, and Hinton 1986), the representation of information is distributed over much of the network in a specialized sense—the state vector of the network weights is the representation. Third is the softness of constraints. Because the network contains a large number of units, each bearing a small responsibility for the task, and because of the continuity of the space over which the weights take values, the output of the network tends to be more or less smooth over the input space. Fourth is learning. Because of a belief that connectionist schemes are particularly good at learning, often an accompanying belief exists that connectionism offers a way to avoid programming an AI system and let learning processes discover all the needed representations.

The properties of parallelism and distribution have attracted adherents who feel that human memory has a holistic character—much like a hologram—and consequently react negatively to discrete symbol-processing theories because they compute the needed information from constituent parts and their relations. Dreyfus (1979), for example, argues that pattern recognition in humans does not proceed by combining evidence about constituent features of a pattern but, rather, uses a holistic process. Thus, Dreyfus looks to connectionism as vindication of his long-standing criticism of symbolic theories. Connectionism is said to perform direct recognition, and symbolicism performs recognition by sequentially computing intermediate representations.

***Connectionism does not offer a royal road to learning***

These characteristics are especially attractive to those who believe that AI must be based more on brainlike architectures, even though within the connectionist camp, a wide divergence is present about the degree to which directly modeling the brain is considered appropriate. Although some of the theories explicitly attempt to produce neural-level computational structures, others propose an intermediate subsymbolic level between the symbolic and neural levels (Smolensky 1988); still others offer connectionism as a computational method that operates in the symbolic-level representation itself (Feldman and Ballard 1982). The essential idea uniting these theories is that the totality of connections defines the information content rather than the representation of information as a symbol structure.

## Is Connectionism Merely an Implementation Theory?

Several arguments have been made that connectionism can, at best, provide possible implementations for symbolic theories. According to one, continuous functions are thought to be the alternative to discrete symbols; because they can be approximated to an arbitrary degree of precision, it is argued that one need only consider symbolic solutions. Another argument is that connectionist architectures are thought to be the implementation medium for symbolic theories, much as the computer hardware is the implementation medium for software. In the subsection entitled Symbolic and Nonsymbolic Representations, we consider and reject these arguments. We show that symbolic and nonsymbolic solutions can be alternative theories in the sense that they can make different representational commitments.

Yet another argument is based on a consideration of the properties of high-level thought, in particular, language and problem-solving behavior. Connectionism by itself does not have the constructs for capturing these properties, the argument runs, so, at best, it can only be a way to implement the higher-level functions. We discuss this point and related issues in Roles of Symbolic and Connectionist Processes.

Having granted that connectionism can make a theoretical difference, we now argue the difference connectionism makes is relatively small to the practice of most of AI.

## Information-Processing Abstractions

Some proponents of connectionism claim that although solutions in the symbolic paradigm are composed of constituents, connectionist solutions are holistic. Composition, in this argument, is taken to be, intrinsically, a symbolic process. Certainly, for some simple problems, connectionist solutions exist with this holistic character. Some connectionist solutions to character recognition, for example, directly map from pixels to characters and cannot be explained as composing evidence about the features, such as closed curves, lines, and their relations. Character recognition by template matching, a nonsymbolic though not a connectionist solution, is another example whose information processing cannot be explained as feature composition. However, as problems get more complex, the advantages of modularization and composition are as important for connectionist approaches as they are for symbolic computation.

Let us consider word recognition, a problem area that has attracted significant attention in connectionist literature. In particular, consider recognition of the word TAKE as discussed by McClelland and Rumelhart (1981). A featureless connectionist solution similar to the one for individual characters can be imagined, but a more natural solution is one that in some sense composes the evidence about individual characters into a recognition of the word TAKE (see sidebar 2). In fact, the connectionist solution that McClelland and Rumelhart describe has a natural interpretation in these terms. Just because the word recognition is done by composition does not mean that each of the characters is explicitly recognized as part of the procedure or that the evidence is added together in a step-by-step, temporal sequence.

Why is such a compositional solution more natural? Reusability of parts, reduction in learning complexity, and greater robustness as a result of intermediate evidence are the major computational advantages of modularization. If the reader doesn't see the power of modularization for word recognition, consider sentence recognition: If one were to go directly from pixels to sentences, without in some sense going through words, the number of recognizers and their complexity would have to be quite large even for sentences of bounded length. Composition is a powerful aid against complexity, whether the underlying system is connectionist or symbolic (Simon 1969). Of course, connectionism provides one style for composition, and symbolic methods provide another, each with its own signature in terms of the performance details.

These examples also raise questions about the degree to which connectionist representations can be distributed. For complex tasks, information is, in fact, localized into portions of the network. Again, in the network for recognition of the word TAKE, physically local subnets can be identified, each corresponding to one of the characters. Thus, hopes for almost holographically distributed representations are bound to be unrealistic.

## The Information-Processing Level

Marr (1982) originated the method of information-processing analysis as a way to separate the essential elements of a theory from implementation-level commitments. He proposed that the following methodology be adopted for this purpose. First, identify an information-processing function with a clear specification about what kind of information is available for the func-

## A Connectionist Solution to Word Recognition

For an illustration of how connectionist networks work, let us consider the model proposed by McClelland and Rumelhart (1981) for the perception of letters of visually presented words. Our description of their model closely follows McClelland, Rumelhart, and Hinton (1986). Their model contains four sets of detectors for the four-letter input words, with a set of units assigned to detect visual features in each of the four different letter positions. The feature-detecting units for one of the letter positions are shown in figure 2. There are four sets of detectors for the letters themselves, and one set for the words. Each unit in the network has an activation value that corresponds to the strength of the hypothesis which states what the unit stands for is present in the input. The connections between the units in the network are such that if two units are mutually consistent—in the way that the letter T in the first position is consistent with the word TAKE—then the activation of one unit tends to support the activation of the other. Similarly, if two hypotheses are mutually inconsistent, then the corresponding units tend to inhibit each other.

Let us consider what happens when a familiar stimulus under degraded conditions is presented to this network. Let us suppose that the input display consists of the letters T, A, and K fully visible and enough of the fourth letter to rule out all letters but E or F. Initially, the activations of all units are set at or below zero. When the display is presented, the activations of detectors for features present in each letter position grow above zero. In the first three positions, T, A, and K are unambiguously activated. For the fourth position, the activations of the detectors for E and F start growing as the feature detectors below them are activated. As these detectors become active, they and the detectors for T, A, and K start to activate detectors for words that have these letters in them. A number of words might be partially consistent with the active letters, but only TAKE matches the active letters in all positions. As a result, TAKE becomes more active than any other word and inhibits other words, thereby successfully dominating the pattern of activation among the word units. As TAKE grows in strength, it sends feedback to the letter level, reinforcing the activations of T, A, K, and E. This feedback gives E the upper hand on F in the fourth position, and eventually the stronger activation of the E detector dominates the pattern of activation, suppressing the F detector completely.

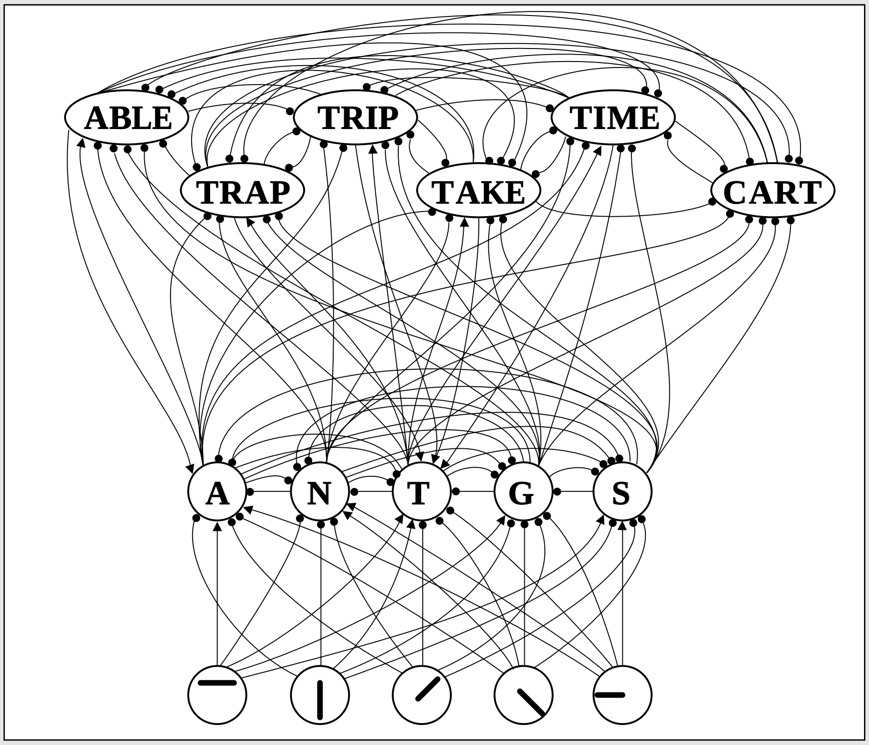


*Figure 2. Connectionist Network for Word Recognition.*

*From "An Interactive Model of Context Effects in Letter Perception: Part 1, An Account of Basic Findings" by J. L. McClelland and D. E. Rumelhart.* Psychological Review *88:380.*

*Photo courtesy of American Psychological Association, copyright 1981. Reprinted by permission.*

tion as input and what kind of information needs to be made available as output. Then, specify a particular information-processing theory for achieving this function by stating what kinds of information need to be represented at various processing stages. Actual algorithms can then be proposed to carry out the information-processing theory. These algorithms make additional representational commitments. In the case of vision, for example, Marr specified that one of the functions is to take image intensities in a retinal image as input and produce as output a three-dimensional shape description of the objects in the scene. His theory of how this function is achieved in the visual system is that three distinct kinds of information need to be generated: first, from the image intensities, a primal sketch of significant intensity changes—a kind of edge description of the scene—is generated. Second, a description of the objects' surfaces and their orientation—what he called a 2-1/2 -dimensional sketch—is produced from the primal sketch. Third, a three-dimensional shape description is generated. Even though Marr talked in the language of algorithms as the way to realize the information-processing theory, in principle, there is no reason why appropriate parts of the realization cannot be done connectionistically.

Information-processing abstractions constitute the content of much AI theory formation. In the recognition of the word TAKE, for example, the information-processing abstractions in which the theory of word recognition was couched evidenced the presence of individual characters. The difference between the recognition schemes in the symbolic and connectionist paradigms is in how the evidence is represented. In the symbolic paradigm, it is represented with labeled symbols, which permit abstract rules of composition to be invoked and instantiated. In the connectionist paradigm, evidence is represented more directly and affects the processing without undergoing any interpretive process. Describing a piece of a network as evidence about a character is a design and explanatory stance and is not necessarily part of the actual information processing in connectionist networks.

As connectionist structures are built to handle increasingly complex phenomena, they will have to incorporate their own versions of modularity and composition. Already we saw such modularity in the moderately complex word-recognition example. When—and if—we finally have connectionist implementations solving a variety of high-level cognitive problems (say, natural language understanding or problem solving and planning), the design of such systems will have an enormous amount in common with the corresponding symbolic theories. This commonness will be at the level of information-processing abstractions that both classes of theories would need to embody. In fact, the contributions of many of the nominally symbolic theories in AI are really at the level of the information-processing abstractions to which they make a commitment and do not rely on the fact that they were implemented in a symbolic structure. Symbols have been often used to stand for abstractions that need to be captured one way or another. The hard work of theory making in AI will always remain at the level of proposing the right information-processing level abstractions because these abstractions provide the content of the representations. The decisions about which of the information-processing transformations are best done using connectionist networks and which using symbolic algorithms can properly follow once the information-processing–level specification is given. Thus, the connectionist and symbolic approaches are realizations of the information-processing–level description, which is more abstract than either realization.

***Radical connectionism, similar to radical symbolicism, seems to demand all of cognition as its domain, and we argue that this demand cannot be conceded***

### Architecture-Independent and -Dependent Decompositions

We argued earlier that for a given function, the symbolic and nonsymbolic approaches might make rather different representational commitments. We also just argued, seemingly paradoxically, that for complex functions the two theories converge in their representational commitments. To clarify, think of two stages in the decomposition of the function: architecture independent and architecture dependent. The architecture-independent stage is an information-processing theory that can be realized by either symbolic or connectionist architectures. In either case, further architecture-dependent decomposition decisions need to be made. In particular, connectionist architectures offer some elementary functions that are rather different from those assumed in traditional symbolic approaches. Simple functions such as multiplication are so close to the architecture level that we only saw the differences between the representational commitments of the algorithmic and slide rule solutions. However, the word-recognition problem is sufficiently removed from the architectural level that we saw information-processing–level similarities between symbolic and connectionist solutions.

Where the architecture-independent information-processing theory stops and the architecture-dependent realization starts is not clear. It is an empirical issue, partly related to the primitive functions that can be computed in a particular architecture. The further away a problem is from the architectures' primitive functions, the more important the architecture-independent decompositions. The final performance will, of course, have features that are characteristic of the architecture, such as the softness of constraints for connectionist architectures.

### Learning to the Rescue?

What if connectionism can provide learning mechanisms such that a network starts without representing any information-processing abstractions and learns to perform the task in a reasonable amount of time; that is, it discovers the needed abstractions by learning? In this case, connectionism can sidestep pretty much all the representational problems and dismiss them as the bane of symbolicism. The fundamental problem of complex learning is the credit-assignment problem, that is, deciding what part of the system is responsible for either the correct or the incorrect performance in a case so that the learner knows how to change the system's structure. Abstractly, the range of variation of the system's structure can be represented as a multidimensional space of parameters, and learning can be viewed as a search in this space for a region that corresponds to the right structure of the system. The more complex the learning task, the more vast the space in which to do the search. Thus, learning the correct set of parameters by search methods that do not have a powerful notion of credit assignment would work in small search spaces but would be computationally prohibitive for realistic problems. Does connectionism have a solution to this problem?

In connectionist schemes, a significant part of the abstractions needed are built into the architecture in the choice of input, feedback directions, allocation of subnetworks, and semantics that underlie the choice of layers for the connectionist schemes. Thus, the input and the initial configuration incorporate a sufficiently large part of the abstractions needed that what is left to be discovered by the learning algorithms, although nontrivial, is proportionately small. The initial configuration decomposes the search space for learning in such a way that the search problem is much smaller in size. In fact, the space is sufficiently small that statistical associations do the trick. As long as the space to be searched is not large, as long as there are no local minima, and as long as there are enough trials, hill climbing can find the appropriate region in the space.

Again, the recognition scheme for TAKE well illustrates this point. In the connectionist scheme cited earlier, the decisions about which subnet is going to be largely responsible for T, which for A, and so on, as well as how the feedback is going to be directed, are all essentially made by the experimenter before any learning starts. The underlying information-processing theory is that evidence about individual characters is going to be formed directly from the pixel level, but recognition of TA is done by combining information about the presence of T and A as well as their joint likelihood. The degree to which the evidence about them is combined is determined by the learning algorithm and the examples. In setting up the initial configuration, the designer is actually programming the architecture to reflect the information-processing theory of recognizing the word. An alternate theory for word recognition, say, one that is more holistic than this theory (that is, one that learns the entire word directly from the pixels), has a different initial configuration. Of course, because of a lack of guidance from the architecture about localizing the search during learning, such a network takes much longer to learn the word. This is precisely the point: The designer recognized this and set up the configuration so that learning can occur in a reasonable time. Thus, although the connectionist schemes for word recognition still make a useful performance point, a significant part of the leverage still comes from the information-processing abstractions with which the designer started.

Additionally, the system that results after learning has a natural interpretation: The learning process can be interpreted as having successfully searched the space for those additional abstractions which are needed to solve the problem. Thus, connectionism is one way to map from one set of abstractions to a more structured set of abstractions. Most of the representational issues remain, whether one adopts connectionism for such mappings. Interesting learning theories in the symbolic framework can also be interpreted as starting with a strong set of abstractions to which the learning process adds sufficient new abstractions to solve the task.

Of course, in human learning, although some of the necessary abstractions are programmed in at various times through explicit instruction, a large amount of learning takes place without any designer intervention in setting up the learning structure. However, there is no reason to believe that humans start with a structure- and abstraction-free initial configuration. In fact, to account for the power of human learning, the initial configurations that a child starts with need to contain complex and intricate representations sufficient to support the learning process in a computationally efficient way. One cannot avoid the specification of appropriate initial structures and still get complex learning at different levels of description to take place in less than evolutionary or geologic time. That is, connectionism does not offer a royal road to learning.

## Roles of Symbolic and Connectionist Processes

In the study of connectionist and symbolic processes, three distinctions can be identified as sharing close affinity. The first is the distinction between macrophenomena and microphenomena of intelligence. The second is the distinction between processes that leave markings that last over time and intuitive or subconscious phenomena occurring in an instant. The last is the distinction between symbolic and connectionist processes. These three distinctions need to be unpacked a bit to see what can be allocated to what processes.

Rumelhart, McClelland, and the PDP Research Group (1986) use the term "micro-" in the subtitle of their book to indicate that the connectionist theories are concerned with the fine details of intelligent processes. A duration of 50–100 milliseconds has often been suggested as the size of the temporal grain for processes at the micro level. However, certain aspects of human cognitive behavior actually evolve over time on a scale of seconds, if not minutes, and have a clear temporal ordering of the major behavioral states. These processes can be termed macrophenomena of intelligence.

Perceptual processes such as face recognition and cognitive processes such as being reminded are examples of microphenomena. As an example of macrophenomena, consider the goal-directed problem-solving activity that a system such as General Problem Solver (GPS) (Newell and Simon 1972) tries to model. The agent is seen to have a goal at a certain instant, to set up a subgoal at another instant, and so on. Within this problem-solving behavior, the selection of an appropriate operator, which is typically modeled in GPS implementations as a retrieval algorithm from the Table of Connections, could be a micro behavior. Many phenomena of language and reasoning have a large macro component.

Neither traditional symbolic computationalism nor radical connectionism has much use for this distinction because all the phenomena of intelligence, micro and macro, are meant to come under their particular purview. We want to present the case for a division of responsibility between connectionism and symbolic computationalism in accounting for the phenomena of interest.

Let us take the macro, conscious-level phenomena first. It seems inescapable that macrophenomena have a high degree of symbolic and algorithmic content, whatever one's beliefs about the formal nature of microphenomena might be. (See Pylyshyn [1984] for compelling arguments in this regard.) How much of language and other aspects of thought require symbol structures can be a matter of debate, but certainly, logical reasoning and goal-directed problem solving such as with GPS are two examples of such behavior.

What follows is a range of phenomena that seem to have a micro, below-conscious character, but whose formal requirements nevertheless place them largely on the symbolic, algorithmic side. For example, natural language sentence comprehension, which generally takes place instantly or as a reflex, nevertheless seems to require such a formal structure. Fodor and Pylyshyn (1988) argue that much of thought has the properties of productivity and "systematicity." *Productivity* refers to a potentially unbounded recursive combination of thought that is presumed in human intelligence. *Systematicity* refers to the capability of combining thoughts in ways that require abstract representation of underlying forms. Fodor and Pylyshyn argue that we need symbolic computations, with their capacity for abstract forms and algorithms, to realize these properties.

Thus, macrophenomena and significant parts of microphenomena not only need the appropriate information-processing abstractions available, but at least parts of them need the abstractions encoded and manipulated symbolically. Whether the symbolic view needs to be adopted for implementation of the other parts is the next question. If any of them can be identified with microphenomena that have a particularly appealing connectionist realization, then one might have an interesting division of responsibility.

Are there such microphenomena? The symbolic paradigm has traditionally assumed that the symbolic, algorithmic character of the macrophenomena also characterizes the inner workings of the cognitive processor that generates the macrophenomena. Connectionism clearly challenges this assumption. Radical connectionism, similar to radical symbolicism, seems to demand all of cognition as its domain, and we argue that this demand cannot be conceded. Nevertheless, the architectures in the connectionist mold offer some elementary functions which are rather different from those assumed in the traditional symbolic paradigm. In particular, certain kinds of retrieval and matching operations and low-level parameter learning are especially appropriate elementary functions for which connectionism offers methods with attractive properties. Thus, a number of investigators in macro AI correctly feel the attraction of connectionist approaches for some parts of their theory formation, the parts where one or more of such elementary functions seem necessary. In a theory such as GPS, for example, the retrieval of the appropriate operators has traditionally been implemented in a symbolic framework, but a connectionist realization of this retrieval seems to have useful properties. (As another example, Anderson and Mozer [1981] propose a model of retrieval using spreading activation [which has a connectionist ring to it], where the objects of retrieval still have significant symbolic content to them. Also, sidebar 3 on connectionism and word pronunciation is an example of connectionism being used within a largely symbolic framework.) Connectionism and symbolicism have different but overlapping domains. A complete theory that integrates these domains along the lines suggested here can be a source for powerful explanations of the total range of the phenomena of intelligence.

The proposed division of responsibility echoes in the proposal in Smolensky (1988) that connectionism operates at a lower level than the symbolic, a level he calls subsymbolic. He also posits the existence of a conscious processor and an intuitive processor. The connectionist proposals are meant to apply directly to the intuitive processor. The conscious processor can have algorithmic properties, according to Smolensky, but still a large part of the information-processing activities that were traditionally attributed to symbolic architectures really belong in the intuitive processor.

Nevertheless, the style of integration proposed leaves a number of problems to be solved. The first problem is how to get the symbolic properties of behavior at or near the level of consciousness out of the connectionist architecture. Additionally, the theory cannot relegate conscious thought to the status of an epiphenomenon. We know that the phenomena of consciousness have a causal interaction with the behavior of the intuitive processor. What we consciously learn and think affects our unconscious behavior slowly but surely, and vice versa. What is conscious and willful today becomes unconscious tomorrow. All this raises a complex constraint for connectionism: It now needs to provide some sort of continuity of representation and process so that this interaction can take place smoothly.

Our account does not merely relegate connectionism to an implementation status similar to relation between computer software and hardware. Because the primitive functions that connectionism delivers are quite different from those assumed in the symbolic framework, their availability changes theory making for the overall symbolic process in a fundamental way. The theory now has to decompose the symbolic process to take special advantage of the power of connectionist primitives. For example, problem-solving theories of expert behavior might radically differ if retrieval were to be a large component of such theories, making problem solving by retrieval of past cases and modification of their solutions an especially dominant component, as in case-based reasoning.

It is important to note that this proposal of the division of responsibility does not mean abandoning the role of information-processing abstractions we have been arguing for. One should be careful about putting too much faith in connectionist mechanisms. As we stated earlier, the power for these operations is going to come from appropriate encodings that get represented connectionistically. Thus, although memory retrieval might have interesting connectionist components, the basic problem is still to find the principles by which episodes are indexed and stored, except that now one might be open to these encodings being represented connectionistically.

### A Connectionist Solution to the Pronunciation Problem

For an example of a connectionist network that is not merely an implementation of a symbolic algorithm but also benefits from using appropriate information-processing abstractions, consider the PRO system of Lehnert (1987).

The task is to use a large case base of words and their pronunciations to learn to pronounce novel words. Cases are presented as letter string and phoneme pairs. Thus, the pronunciation of the word showtime results in the sequence of pairs (SH/sh, OW/o, T/t, I/i, ME/m). This sequence is split into triplets, for example, (OW/o, T/t, I/i, for training. The number of occurrences of each triplet during training is counted.

When a query is presented to PRO, it generates all possible hypothesis sequences it can associate with the word. Note that this output does not usually contain all possible segmentations of the input word because most substrings are not associated with hypotheses encountered in training. These hypotheses are linked into a network, with supporting connections between hypotheses that correspond to a particular segmentation of the word and inhibitory connections between hypotheses which represent different uses for the same input letter. A node that Lehnert refers to as a "context node" is added to the network wherever three consecutive hypotheses correspond to one of the triples encountered during training. The activation levels of the context nodes are computed based on the number of occurrences of this triplet. A standard relaxation algorithm is then applied to the network to decide which pronunciation to prefer.

This solution shows the same fuzziness as connectionist solutions inasmuch as the recognition is based on patterns that emerge from the corpus of cases. However, the learning is not done in a standard connectionist fashion. The power of a learning scheme comes from its capability to successfully solve the credit-assignment problem. PRO makes a statement about the credit-assignment problem by using frequency of hypothesis sequences as the basis of learning. This approach makes learning much faster because the necessary abstractions are already present in the system, and credit assignment is focused. The power of this method for assigning credit comes from the appropriate information-processing abstractions of phonetic hypotheses.

At the information-processing level, this theory states that the appropriate way to decide how to pronounce a word is to break it into groups of letters which correspond to phonetic hypotheses rather than into the obvious units of individual letters. Furthermore, frequencies of phonetic hypothesis sequences in a case base can distinguish which hypotheses to use. At the architecture level are the specific relaxation algorithm and the context nodes. The success of this method comes from the information-processing abstractions, and the fuzziness of the solution comes from the connectionist architecture.

Finally, we want to address the comment by Rumelhart et al. (1986) that symbolic theories are really explanatory approximations of theories which are connectionist at a deeper level. As an example, they suggest that a schema or a frame is not really explicitly represented as such but is constructed, as needed, from general connectionist representations. This suggestion seems plausible but does not mean that schema theory is only a macroapproximation. Schema, in the sense of being an information-processing abstraction needed for certain macrophenomena, is a legitimate conceptual construct for which connectionist architectures offer a particularly interesting realization. It is not that connectionist structures are the reality and that symbolic accounts provide an approximate explanation; rather, it is the information-processing abstractions which contain a large portion of the explanatory power.

## Conclusion

What impact will connectionism have on AI in general? Much of AI research, except where microphenomena dominate and symbolic AI is simply too hard edged in its performance, will and should remain largely unaffected by connectionism for two reasons. First, most of the work is in discovering the information-processing theory of a phenomenon in the first place. The further the task-level description is from the phenomenon at the raw architecture level, the more common are the representational issues between the connectionist and symbolic approaches. Second, none of the connectionist arguments or empirical results show that the symbolic, algorithmic character of thought is a mistaken hypothesis, purely epiphenomenal, or simply irrelevant.

Our arguments for and against connectionist notions are not really specific to any particular scheme. They are intended to apply to nonsymbolic approaches in general, including the approaches of Hopfield and Tank (1985). The work of Reeke and Edelman (1988) challenges any form of representationalism, which requires a separate answer. Within representationalist theories, however, it seems that we need to find a way to deal with three constraints on architectures for mental phenomena: (1) A large part of theory making in AI has to do with the content of mental representations. We call them the information-processing abstractions. (2) Whatever one’s position on the nature of representations below conscious processes, it is clear that processes at or close to this conscious level are intimately connected to language and knowledge and, thus, have a large discrete symbolic content. (3) The connectionist ideas on representation suggest how nonsymbolic representations and processes can provide the medium in which thought resides.

From the viewpoint of computer science, connectionism has done a useful service in refocusing attention on alternative models of computation. However, for much of AI and cognition, the supposed battle between connectionism and symbolicism is mere shadowboxing. Neither of the theories explains or accounts for all intelligence or cognition. The task of building a natural language understanding system is not even remotely complete just because we have a bucket of connectionist units and weights or, equally, a universal Turing machine in front of us. Sure, it is nice to know they both provide a certain kind of universality (if, in fact, connectionist architectures do), but beyond this, it is time to make theories at a different level of description altogether.

As said in Chandrasekaran (1986) in a slightly different context, “There has been an ongoing search for the ‘holy grail’ of a uniform mechanism that will explain and produce intelligence. This desire has resulted in a number of candidate mechanisms—from perceptrons of the 1960s through first-order predicate calculus to rules and frames—to satisfy this need.” Intelligence is not a product of any one mechanism, whether at the connectionist or rule level. Reductionism, either of the connectionist or symbolic style, misstates where the power of intelligence as a phenomenon is coming from. Its power is a result of cooperation between different mechanisms and representations at different levels of description.

## Acknowledgments

We gratefully acknowledge the support of the Defense Advanced Research Projects Agency, RADC contract F30602-85-C-0010, the Air Force Office of Scientific Research, grant 87-0090, and the National Science Foundation for the graduate fellowship (Allemang) granted during the preparation of this article.

## References

Anderson, J. R., and Mozer, M. C. 1981. Categorization and Selective Neurons. In *Parallel Models of Associative Memory,* eds. G. E. Hinton and J. R. Anderson, 213–236. Hillsdale, N.J.: Lawrence Erlbaum.

Bruner, J. S. 1957. On Perceptual Readiness. *Psychological Review* 64:123–152.

Chandrasekaran, B. 1986. Generic Tasks in Knowledge-Based Reasoning: High-Level Building Blocks for Expert System Design. *IEEE Expert* 1(3): 23–30.

Dreyfus, H. L. 1979. *What Computers Can’t Do.* New York: Harper & Row.

Feldman, J. A., and Ballard, D. H. 1982. Connectionist Models and Their Properties. *Cognitive Science 6*:205–254.

Fodor, J. A., and Pylyshyn, Z. W. 1988. Connectionism and Cognitive Architecture: A Critical Analysis. *Cognition* 28:3–71.

Gibson, J. J. 1950. *The Perception of the Visual World.* Boston: Houghton-Mifflin.

Hopfield, J. J., and Tank, D. W. 1985. Neural Computation of Decisions in Optimization Problems. *Biological Cybernetics* 52:141–152.

Lehnert, W. G. 1987. Case-Based Problem Solving with a Large Knowledge Base of Learned Cases. In *Proceedings of the Sixth National Conference on Artificial Intelligence, 3*01–306. Menlo Park, Calif.: American Association for Artificial Intelligence.

McClelland, J. L., and Rumelhart, D. E. 1981. An Interactive Activation Model of Context Effects in Letter Perception: Part 1. An Account of Basic findings. *Psychological Review 8*8:375-407.

McClelland, J. L.; Rumelhart, D. E.; and Hinton, G. E. 1986. The Appeal of Parallel Distributed Processing. In *Parallel Distributed Processing,* vol. 1, eds. D. E. Rumelhart, J. L. McClelland, and the PDP Research Group. Cambridge, Mass.: MIT Press/Bradford.

McClelland, J. L.; Rumelhart, D. E.; and the PDP Research Group, eds. 1986. P*arallel Distributed Processing,* 2 vols. Cambridge, Mass.: MIT Press/Bradford.

McCulloch, W. S., and Pitts, W. 1943. A Logical Calculus of the Ideas Immanent in Nervous Activity. *Bulletin of Mathematical Biophysics* 5:115–137.

Marr, D. 1982. *Vision.* San Francisco: Freeman.

Newell, A. 1980. Physical Symbol Systems. *Cognitive Science* 4:135–183.

Newell, A., and Simon, H. A. 1972. *Human Problem Solving.* Englewood Cliffs, N.J.: Prentice-Hall.

Pylyshyn, Z. W. 1984. *Computation and Cognition: Towards a Foundation for Cognitive Science. C*ambridge, Mass.: MIT Press.

Reeke, G. N., Jr., and Edelman, G. M. 1988. Real Brains and Artificial Intelligence. *Daedalus* 117(1): 143–173.

Rosenblatt, F. 1962. *Principles of Neurodynamics.* New York: Spartan.

Rumelhart, D. E.; Smolensky, P.; McClelland, J. L.; and Hinton, G. E. 1986. Schemata and Sequential Thought Processes in PDP Models. In *Parallel Distributed Processing: Explorations in the Microstructure of Cognition,* vol. 2, eds. J. L. McClelland, D. E. Rumelhart, and the PDP Research Group. Cambridge, Mass.: MIT Press/Bradford.

Simon, H. A. 1969. *The Sciences of the Artificial.* Cambridge, Mass.: MIT Press.

Smolensky, P. 1988. On the Proper Treatment of Connectionism. *Behavioral and Brain Sciences,* 11(1):1–23.